Actor

Main

CourseRegistiration
Simulation

Rand
Stud

"Course registiration simulation starting."

simulation.startSimulation()

generateStudents(input)

createRandomStudent

createStudentTr

appendCoursesAtSeme

registerRequestedCo

simulateGrades(re

Participants: (Random Student), Transcript, Course, TakenCourse

numberOfStudents>getNumberofStudents

new Student(ID,FName,LName,advisor)

...ranscript(semester,student)

getTakenCourses()

if(takenCoursestatus=="failed")

requestedCourses.add(course)

getNotRegisteredCourse()

if(notRegisteredCourse.getSemester().equals(currentSemster)

requestedCourses.add(course)

...ester(semester,requestedCourses)

...ourses(requestedCourse,transcript)

if(course.getPrequsite()==null || prerequisiteInTranscript.getTakenCourseStatus().equals("Passed")

requestedCourses.add(course)

else

transcript.addSelectionProblem(course)

...egisteredCourses)

...ddTakenCourse(simulatedGrades.get())

```
SelectionProblem                    Student              Course
                                                         Registiration
                                                         System


                                                    InputJson input
```

InputJson input

getRandomFirstName(); getRandomLastName()

```
CourseStatistics          outputJson          inputJson
```

Person

Advisor

numberOfStudents+" Random student created"

Arraylist<Student>

"Students registiration completed"

"Students saved into Json files"

"Course registiration simulation ended"

addTakenCourse(simulatedGrades.get())

setTranscript(transcript)

addRequestedCourses(course)

registerStudent(Student)

out

output.saveCourseSta

if(course.getSemester().equals

!prerequisiteInTranscript.getTakenCo

<1 || coursesTa

getCourseStatistics().incrPrerequisiteProblemCount(); co

addSelectionProblem(1,course,description)

transcript.addTakenCourse(course,0,"Current")

put.saveStudent(student)

tistic(input.getCourses(),input.getSemester())

s(semester) **||** prerequisiteInTranscript==null ||
ourseStatus().equals("Passed") **||** course.getQuota()
ken>MAX_COURSE_COUNT)

urse.getCourseStatistics().incrRegistrationFailureCount()